

Biologically inspired kinematic synergies enable linear balance control of a humanoid robot

Helmut Hauser · Gerhard Neumann ·
Auke J. Ijspeert · Wolfgang Maass

Received: 22 November 2010 / Accepted: 6 April 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract Despite many efforts, balance control of humanoid robots in the presence of unforeseen external or internal forces has remained an unsolved problem. The difficulty of this problem is a consequence of the high dimensionality of the action space of a humanoid robot, due to its large number of degrees of freedom (joints), and of non-linearities in its kinematic chains. Biped biological organisms face similar difficulties, but have nevertheless solved this problem. Experimental data reveal that many biological organisms reduce the high dimensionality of their action space by generating movements through linear superposition of a rather small number of stereotypical combinations of simultaneous movements of many joints, to which we refer as kinematic synergies in this paper. We show that by constructing two suitable non-linear kinematic synergies for the lower part of the body of a humanoid robot, balance control can in fact be reduced to a linear control problem, at least in the case of relatively slow movements. We demonstrate for a variety of tasks that the humanoid robot HOAP-2 acquires through this approach the capability to balance dynamically against unforeseen disturbances that may arise from external forces or from manipulating unknown loads.

Electronic supplementary material The online version of this article (doi:[10.1007/s00422-011-0430-1](https://doi.org/10.1007/s00422-011-0430-1)) contains supplementary material, which is available to authorized users.

H. Hauser (✉) · G. Neumann · W. Maass
Institute for Theoretical Computer Science,
Graz University of Technology, 8010 Graz, Austria
e-mail: helmut.hauser@igi.tugraz.at

A. J. Ijspeert
BIOROB, Biorobotics Laboratory,
École Polytechnique Fédérale de Lausanne,
School of Engineering, 1015 Lausanne, Switzerland

Keywords Kinematic synergies · Humanoid robot · Balance control · Biologically inspired · Motion primitives

1 Introduction

The balance control problem of humanoid robots is known to be hard to solve due to the high dimensionality of their action space (since many degrees of freedoms, i.e., joints, are involved) and the non-linearities inherent to any kinematic chain. Because of the importance of finding solutions to this problem, quite a bit of effort has already been invested and many approaches from different research areas have been proposed.

A first step was made by [Vukobratović and Borovac \(2004\)](#) by introducing the Zero Moment Point (ZMP) criterion. It simplifies the high dimensional problem by reducing all acting forces above the foot (in the case of single support, i.e., contact with the ground with only one foot) to one single force ([Vukobratović and Borovac 2004](#)). Due to physical interaction between foot and ground, we get as a result of Newton's 3rd law (i.e., action–reaction), at the point where this resulting force acts, a so-called ground reaction force with opposite sign. The two-dimensional point (called ZMP) on the ground, where this resulting force acts, can then be used to characterize the dynamic state of the robot: if the ZMP lies within the support polygon of the robot, the state of the robot is called dynamically stable. This so called “ZMP stability criterion” reduces the problem of stability to coordinate the limbs of the robot (i.e., apply appropriate torques through their servos) in such a way, that the ZMP stays within the support polygon¹.

¹ The robot could also change the size of the support polygon by, for example, hold on to something. For a discussion of different control strategies in this context, we refer to [Goswami and Kallem \(2004\)](#).

While the ZMP can be calculated analytically, the position of this point can also be measured by pressure sensors (actually measuring the ground reaction force). From this point of view the resulting point is called accordingly Center of Pressure (CoP). As [Goswami \(1999\)](#) demonstrated, the ZMP equals the CoP since they describe the same phenomenon from different points of view. In this paper, we are going to use the name CoP, since we use the pressure sensor information in combination with the support polygon to estimate the state of stability. Since the original ZMP definition has some limitations ([Goswami 1999](#)), other ground reference points have been proposed, for example, the Foot Rotation Indicator (FRI) introduced by [Goswami \(1999\)](#) or the Centroidal Moment Pivot (CMP), just to name two. For a detailed discussion, we refer to [Popović et al. \(2005\)](#).

Other approaches have been proposed that are also based on a reduced model of the robot. For example, the *Inverted Pendulum Model*, introduced by [Kajita et al. \(1992\)](#), has proved to be very useful. It describes the whole robot, under some assumptions, by a linear inverted pendulum and thereby, reduces the number of dimensions. Extensions of this model have also been studied, for example, the Three-Dimensional Inverted Pendulum Model (3D-LIPM) by [Kajita et al. \(2001\)](#) and the Reaction Mass Pendulum (RMP) by [Lee and Goswami \(2007\)](#). Although all these reduced models are useful, still, at the point of implementation one has to find control schemes which map the strategy back into the full dynamic model (as [Lee and Goswami \(2007\)](#) pointed out). Hence, they have difficulties dealing with unknown external perturbations, since these perturbations present a change in the dynamics of the robot.

An alternative approach to balance control is to rely on the static model, i.e., to use the kinematic model and the mass distribution of the robot. By employing a local Jacobian Pseudo-Inverse (JPI) approach on local information, like Resolved Motion Rate Control (RMRC) ([Whitney 1969](#)), the optimal change of the joint angles can be calculated. Some of these frameworks even allow to set priorities among conflicting tasks, e.g., [Baerlocher and Boulic \(1998, 2004\)](#). Accordingly, balancing could be one of these tasks, typically with a high priority. In order to deal with unforeseen perturbations, the setup has to be used inside a feedback control loop, for example as proposed in [Mansard and Chaumette \(2007\)](#). However, a drawback of such an approach is that it calculates online inverse kinematics, which involves computationally expensive matrix inversions.

Other approaches try to solve directly the dynamic equations within constraints, which reflect the border of stability. For example, [Kagami et al. \(2001\)](#) proposed an online balancing scheme by solving a quadratic programming problem. However, the precise dynamic model of the robot is needed in order to apply this approach. Therefore, it has difficulties in situations where the dynamic model of the robot signif-

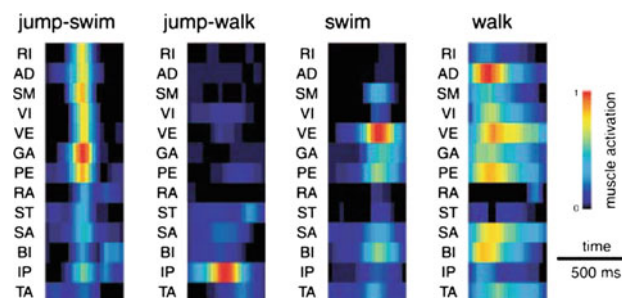


Fig. 1 The figure is a part of a Fig. 4 taken from [d'Avella and Bizzi \(2005\)](#). They present shared and behavior-specific time-varying synergies (i.e., activation of 13 muscles of adult bullfrogs over time). We adopt this concept for robots by defining synergies in the context of time-varying joint angles (compare Fig. 7)

icantly changes due to external unknown forces, for example, introduced by picking up unknown loads or contact with the environment, which are standard situations for humanoid robots working in a human environment.

Biological organisms face similar problems, but, as experimental data suggest, employ a radically different strategy for controlling their movement apparatus with many degrees of freedom (DoF), in particular for balance control. Numerous studies from the Lab of Bizzi at MIT ([Mussa-Ivaldi 1999](#); [d'Avella et al. 2003](#); [d'Avella and Bizzi 2005](#)) have shown that the central nervous systems of a variety of organisms employ a modular architecture for motor control, whereby many different movements (arm movements, walking, jumping, and swimming) can be constructed as largely linear (but non-negative) combinations of a rather small repertoire of movement primitives (also referred to as muscle synergies, or kinematic synergies; we use the latter term in this article). For example, Fig. 1 shows a result from [d'Avella and Bizzi \(2005\)](#). It depicts the underlying non-linear orchestration of 13 muscles activations², i.e. synergies, for different behaviors and synergies, which are shared between different behaviors for bullfrogs. However, such synergies are not only found in rather simple animals, but also in complex biological systems like humans.

For example, recent work on whole-body movements of humans ([Freitas et al. 2006](#); [Tricon et al. 2007](#); [Torres-Oviedo and Ting 2007](#)) show that balance control and other human body movements during standing can be understood as combinations of a small set of stereotypical kinematic synergies (each of them affects several joints). Experiments, where humans were asked to bend their upper trunk, while recording the angles of the ankle, hip and knee, revealed after a Principal Component Analysis (PCA) of these angles, that already the first principle component can explain over 99% of the total angular variance ([Alexandrov et al. 1998](#)). This suggests that a set of muscles (multiple degrees of freedom) are

² obtained by a non-negative matrix factorization algorithm

controlled by a low dimensional (possibly even one-dimensional) variable. Other experiments suggest that this principle of kinematic synergies is present over a wide range of different movements like reaching and grasping (Mason et al. 2001), upper-arm movement (Sabatini 2002), and making a step (Wang et al. 2005). Hence, kinematic synergies seem to present a general strategy biological organisms apply.

In this work, we adopt this concept for real robots in general and for balance control of humanoid robots in particular. Therefore, we consider instead of a time-varying activations of different muscles (as in Fig. 1) time-varying joint angles of different joints. Although there exists an obvious gap between muscles of a biological systems and high torque driven joints we are still able to adopt this concept by defining abstract synergies in mathematical terms, as we have already demonstrated in a preceding conference paper (Hauser et al. 2007). We showed how this basic modular strategy (based on kinematic synergies) can be adapted for balance control of a humanoid robot. The kinematic synergies were calculated offline by an optimization process based only on the *static* model (kinematics and masses) of the robot³. Despite the use of the static model, we could demonstrate that the concept of kinematic synergies, when plugged into a linear control loop, can provide a powerful scheme for *dynamic* balance control. This article presents an extension of the previous work, i.e., Hauser et al. (2007), by following points: (1) We demonstrate that our approach of kinematic synergies is robust to parameter changes of the model of robot. Actually, changes of the static model present a standard situation for biological systems since they grow or even get injured (e.g., losing a leg). (2) Additionally, we show that no special tuning of the controller parameters is needed since the proposed framework works (i.e., balances the robot) within a wide range of these parameters. (3) We demonstrate that the chosen kinematic synergies, originally designed for double support, can also be applied for the case of single support. (4) Finally, we demonstrate that the proposed approach for balance control can be transferred from a simulated humanoid robot without any changes to a real humanoid robot. In the next section, we define kinematic synergies. Section 3 explains how to construct and use kinematic synergies for balance control of the humanoid robot HOAP-2. In Sect. 4, we present a number of experiments with the simulated and the real HOAP-2.

2 Formal definitions of kinematic synergies

In this section, we define kinematic synergies in order to reduce high dimensionality and non-linearities. Typically,

³ This optimization process is closely related to the Jacobian Pseudo-Inverse approaches (Sciavicco and Siciliano 1999), however, the computations are only needed for the (offline) construction of the synergies (and therefore are calculated only once) and not during online control.

humanoid robots have a high number of degrees of freedom (DoF), namely joints. We interpret kinematic synergies (KS) as a way to reduce the DoF by putting a defined set of joints under the regime of one controlling parameter, which we refer to as the KS-parameter s . We define a kinematic synergy as a non-linear mapping Φ of the KS-parameter $s \in \mathbb{R}$ to a fixed number of m degrees of freedom (joints).

Definition 1 A KS is a function $\Phi := \Phi(s)$ which maps the KS-parameter $s \in \mathbb{R}$ onto a m dimensional vector of joint angles $\mathbf{q}^{KS} = \Phi(s)$:

$$\Phi : \mathbb{R} \rightarrow \mathbb{R}^m. \quad (1)$$

The superscript KS denotes the subset of m joints, which are controlled by the KS. The total number of joints of the robot is denoted by n . Further, we define the function φ

$$\varphi : \mathbb{R}^m \rightarrow \mathbb{R}^n \quad (2)$$

to embed the m -dimensional subspace spanned by Φ into the n -dimensional space of all joints of the robot. This embedding copies the angles of all joints affected by Φ and leaves the remaining joints constant.

A KS is typically applied in order to control a low dimensional, or even one-dimensional, variable $y \in \mathbb{R}^l$. In general, the output y depends on all n joint positions $\mathbf{q} \in \mathbb{R}^n$ of the robot and can be described by a non-linear function $\mathbf{f}(\mathbf{q})$

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^l. \quad (3)$$

We want the KS to control the output $y = (\mathbf{f} \circ \varphi \circ \Phi)(s)$, see Fig. 2. In the case of balance control, the function \mathbf{f} represents the non-linear relationship between all joints of the robot and a ground reference point like the CoP. We will use two KS Φ_x and Φ_z for the two-dimensions of the CoP. Therefore, in this particular case each KS is used to control a one-dimensional output ($l = 1$).

Since such a KS affects m degrees of freedom that depend just on a one-dimensional parameter s , we can impose further constraints on the function Φ . A reasonable choice for such a constraint is a linear relationship between the controlling parameter s and its corresponding output y . This reduces non-linearities, inherent to kinematic chains, and hereby

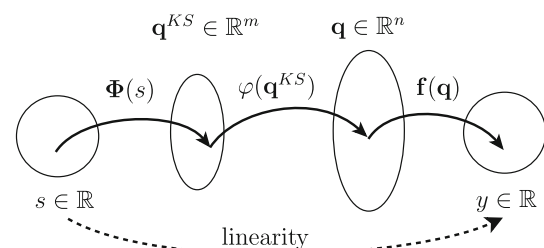


Fig. 2 Scheme for the composition of the functions φ and \mathbf{f} according to (2) and (3) with the kinematic synergy Φ

facilitates controlling and learning. Hence, we are particularly interested in the following type of KS:

Definition 2 A *linearizing kinematic synergy* is a kinematic synergy according to Definition 1, which has a linear relationship between its controlling parameter s and the corresponding (to be controlled) output y

$$y = (\mathbf{f} \circ \varphi \circ \Phi)(s) = k \cdot s, \quad k \in \mathbb{R}. \quad (4)$$

We restrict our attention in this article to such *linearizing KS*, to which we simply refer as KS.

For a better understanding, we provide some additional remarks:

1. As stated above the property of linearity in Def. 2 reduces inherent non-linearities. But Eq. 4 presents a static mapping, and therefore, it will only linearize the static part (linearization at $\dot{\mathbf{q}} = \mathbf{0}, \ddot{\mathbf{q}} = \mathbf{0}$) of the whole dynamic model of the robot. Nevertheless, it will reduce non-linearities in the dynamic regime to some extent too, since the dynamic part is coupled with the static part of the differential equations.
2. The controlled variable y is one-dimensional, but is controlled by $m > 1$ joints. Hence, we have additional redundant degrees of freedom and therefore, we are free to impose additional constraints on the KS. Naturally, the choice will depend on the task for which the KS are constructed. In our case of balance control, we used constraints to assure double support and an upright posture (used in the optimization process described in Sect. 3.1).
3. KSs are calculated offline for each robot (see Sect. 3.1) and subsequently fixed during simulation as well as when used with the real robot. In a biological interpretation, we assume the KSs to be found by evolution.
4. The presented framework was kept as simple as possible. Various extensions, which lead to a better performance for particular tasks, are possible. One could define a two-dimensional kinematic synergy (i.e., $s \in \mathbb{R}^2$ and $y \in \mathbb{R}^2$) or time-varying KSs ($\mathbf{q}^{\text{KS}} = \Phi(s, t)$), which depend on a cyclic movement, for example, to be used in a walking cycle.

3 Using kinematic synergies for balance control of the humanoid robot HOAP-2

In this section, we show in detail how to use kinematic synergies for balance control of the humanoid robot HOAP-2, see Fig. 3a. The robot has $n = 25$ degrees of freedom (rotational joints). Its structure can be seen in Fig. 3b. The goal is to construct KSs for balance control in double support. Therefore, we have to decide (a) what output function \mathbf{f} and output variables y we are going to use, (b) which subset of m joints

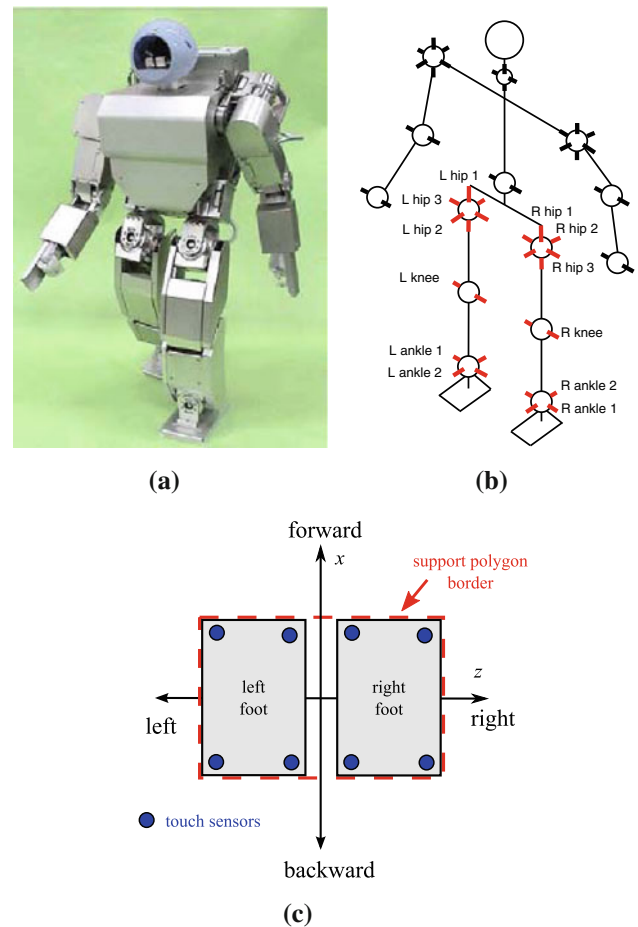


Fig. 3 **a** The real HOAP-2 robot and **b** its schematic structure. The red marked and labeled joint rotation axes are controlled by the kinematic synergies Φ_x and Φ_z . **c** Support polygon on the support surface for the robot, including the touch sensors, which are used to measure the center of pressure (mCoP). Black arrows indicate the x dimension (forward/backward: range 9.5 cm) and z dimension (left/right: range: 14.3 cm) for movements of the CoP

we put under the regime of the KSs, and (c) what additional constraints we are going to apply to construct the KSs:

- (a) For balance control, a natural choice for the function \mathbf{f} is a ground reference point. These points are mathematically defined and can be analytically derived, but in practice, they are estimated via pressure sensors. Therefore, we will denote the reference point measured by the pressure sensors as measured Center of Pressure (mCoP). HOAP-2 has four of such sensors per feet, located at the corners (see Fig. 3c).

Since a KS is defined as a static mapping, we use the static version of the mCoP to construct our KS. In the static case (zero joint velocity $\dot{\mathbf{q}}$ and zero joint acceleration $\ddot{\mathbf{q}}$) the mCoP coincides with the projected Center of Mass (pCoM). Therefore, we chose the pCoM as output function \mathbf{f} . Since the pCoM is a two-dimensional point on the supporting surface, we split it up into its two-dimensions

$y_x = \text{pCoM}_x$ and $y_z = \text{pCoM}_z$ and define two separate KSs, namely Φ_x and Φ_z , in order to control these one-dimensional outputs y_x and y_z .

- (b) Next, we have to decide what joints are placed under the regime of our KSs. A natural choice for balance control is to use all $m = 12$ leg joints (three hip joints, one knee joint, and two ankle joints for both legs). Their corresponding rotational axes are highlighted in red in Fig. 3b. The additional surplus of joints are free to be used for other tasks (grasping, lifting weights, and tracking objects, etc.). Their movements clearly will change the pCoM too, but as we show later in Sect. 4, our approach is able to deal with that in a natural way.
- (c) Finally, we choose some additional constraints (next to the linearity property) for the KSs, which are used for the optimization process described in the next section. Suitable constraints for balance control are to keep the upper body as upright as possible and to maintain double support.

Note that in contrast to our setup for the humanoid robot here, humans use a number of sensory channels for balance control. They incorporate the visual input next to the sensory information from their graviceptive and proprioceptive systems. The latter is the only one that we use in our context and it is mimicked in the HOAP-2 robot in very simple form as touch sensors. However, there exist biologically inspired approaches for balance control that include multiple sensory information. For example, Mahboobin et al. (2008) added the graviceptive sensory information in order to increase the robustness of the balancing. They combined both information streams via a Kalman filter. Another approach has been suggested by Peterka (2009). He demonstrated that a combination of a ZMP compensation scheme (linear feedback loop based on the CoP), a feedback loop based on the ankle angle (proprioceptive sensor), and a feedback loop based on the body tilt angle (graviceptive sensor) is able to increase the balance performance. However, both mentioned approaches have been used so far only to balance a simple, one-dimensional, linear model of the robot (i.e., a linear inverted pendulum), instead of a real robot. An extension of our kinematic synergy approach in order to incorporate multiple sensory information would be in so far interesting, as that the kinematic synergies might be able to close the gap between these simple models and the real robots. However, this would be beyond the scope of this paper.

3.1 Calculating kinematic synergies with inverse kinematics

In this section, we describe the process to obtain the desired KSs in detail. All calculations are based only on the kinematic model of the robot including the mass information

(no dynamical information like the inertia matrices is needed). The KSs were constructed offline and subsequently fixed during control action.

We defined an initial posture \mathbf{q}_{init} (see Fig. 5; posture A). This posture resulted (for the case of a horizontal support surface) in a pCoM at the center of the support polygon. We used a posture with wide-spread arms in order to avoid self collision when moving. The KS-parameters s_x and s_z were rescaled such that the values -1 and $+1$ corresponded to the borders of the support polygon. Therefore, the region of acting without falling was (for the case of a horizontal support surface) $s_x/s_z \in [-1, +1]$ for both dimensions x and z , see red-dashed lines in Fig. 3c. We additionally set the origin of the coordinate system for the pCoM to the center of the support polygon and therefore, the resulting outputs in the initial posture were $\mathbf{f}_x(\mathbf{q}_{\text{init}}) = \mathbf{f}_z(\mathbf{q}_{\text{init}}) = 0$.

We will only describe the procedure for Φ_x . The second kinematic synergy Φ_z was obtained in a similar manner. The KS was implemented as look-up table which maps the KS-parameter $s_x \in [-1, +1]$ to joint angle offsets (with regard to the initial posture)⁴, i.e., $\Delta \mathbf{q}_x = \varphi(\mathbf{q}_x^{\text{KS}}) - \mathbf{q}_{\text{init}}$. Note that the look-up table represents a discretized version of a *linearizing kinematic synergy* as defined in Definition 2. In order to obtain joint angle offsets between the table entries, a linear interpolation was used. We used joint angle offsets instead of absolute joint angles in order to be able to use a linear superposition (as biological data suggest) of both KSs, i.e., $\Delta \mathbf{q} = \Delta \mathbf{q}_x + \Delta \mathbf{q}_z$. Although, the problem is (due to the kinematic chains) non-linear, we will show that a linear superposition is valid for a wide range of postures. The linear superposition allows us to use two separate simple KS, which depend only on a one-dimensional KS-parameter, and which can be constructed independently⁵.

In order to construct the look-up table, we divided the range of the KS-parameter s_x over the support polygon into 80 points. Therefore, the distance between two neighboring points represents $9.5 \text{ cm}/80 \approx 0.12 \text{ cm}$ in the pCoM space, which corresponds to a step of $\Delta s_x = 0.025$ in the KS-parameter space.

The construction of the KS consisted of two alternating optimization steps (see optimization scheme in Fig. 4). Starting from \mathbf{q}_{init} and $s_x = 0$, the first optimization step was used to move the pCoM of the robot to the next point y'_x of the look-up table (located 0.12 cm in x -direction from the origin). In addition, the optimization tried to keep the upper part of the body upright. An inverse kinematics algorithm based on the Jacobian Pseudo-Inverse (JPI) (Sciavicco and Siciliano 1999) was used to calculate the joint movement.

⁴ The function φ is used to project the m -dimensional vector \mathbf{q}_x^{KS} into the n -dimensional space of all joints.

⁵ Without this property, one would have to construct one single KS with a two-dimensional KS-parameter, i.e., $s \in \mathbb{R}^2$.

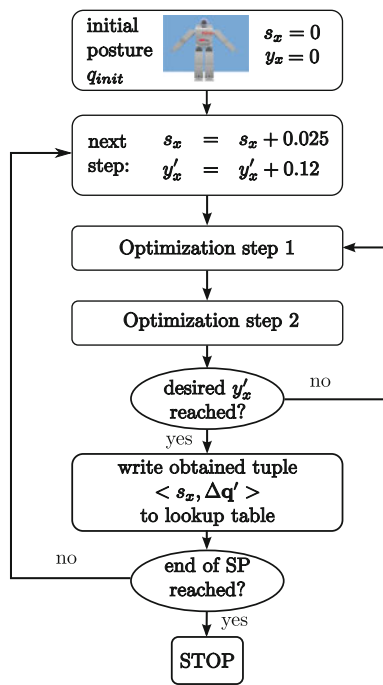


Fig. 4 Scheme of the construction process for the look-up table for the KS Φ_x in the form $\langle s_x, \Delta \mathbf{q} \rangle$. Optimization step 1 moves the pCoM in the desired direction to y'_x , while keeping the trunk in an upright position. Optimization step 2 keeps the feet at the initial positions. The process ends when the end of the support polygon (SP) is reached

Therefore, the applied Jacobian matrix consisted of two $3 \times m$ sub-matrices, the Jacobian for the position of the pCoM and the Jacobian for the rotation of the torso. However, due to the movement calculated by this optimization, the position of the right foot relative to the left foot tended to change. This should be avoided in order to prevent the robot from falling. Therefore, a second JPI optimization step (see Fig. 4) was used to move the right foot back into its original position relative to the left foot. For this optimization, the same Inverse Kinematics algorithm was applied using only the 6 joints of the right leg.

These two previously described steps were iterated until the desired output value y'_x was reached. Subsequently, the joint angle offsets to the initial posture were stored in the look-up table and, now starting from the new joint position, the next entry of the look-up table was calculated. The same process was applied for the opposite direction (i.e., for s_x from 0 to -1). This finally led to a look-up table for the range $s_x \in [-1, +1]$ which mapped the KS-parameter s_x to joint angle offsets.

Figure 5 presents four typical postures for different KS-parameter pairs $[s_x/s_z]$. The center of the figure shows the support polygon (gray area) and the coordinate system of the KS-parameters. The yellow circles (A–D) represent the postures in the KS-parameter space. The corresponding screenshots can be seen in the corners of the figure.

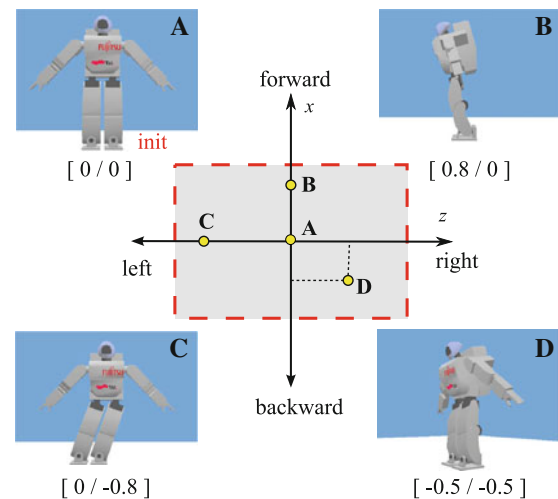


Fig. 5 Typical postures of the simulated HOAP-2 resulting from the KSs Φ_x and Φ_z for different KS-parameters. The center of the figure shows the defined coordinate system for the KS-parameters s_x and s_z . The gray shaded area indicates the support polygon (SP) of our robot standing with both feet on the ground. The red-dashed lines depict the limits of the SP and correspond to the values $s_x = \pm 1.0$ and $s_z = \pm 1.0$. The yellow points show typical postures in the KS-parameter space. Corresponding postures can be seen in the corners (labeled from A to D). The used KS-parameters $[s_x/s_z]$ can be seen below the screenshots. Screenshot A shows the initial posture \mathbf{q}_{init} ($s_x = s_z = 0$ /at the origin) B shows the robot bending forward with $s_x = 0.8$ and $s_z = 0.0$, while in C the robot is bending to the left (with $s_x = 0.0$ and $s_z = -0.8$). Screenshot D presents a combination of both kinematic synergies with $s_x = -0.5$ and $s_z = 0.5$

Figure 6a shows the mapping of the KS-parameter s_x to the outputs $y_x = \text{pCoM}_x$ and $y_z = \text{pCoM}_z$ for the KS Φ_x . We can identify a linear relationship between s_x and y_x , whereas the second output dimension y_z is unaffected by s_x . The same plot for the KS Φ_z is shown in Fig. 6b.

A graphical representation of the joint angle offsets over the range of the KS-parameter spaces (from -1 to $+1$) for the kinematic synergies Φ_x and Φ_z is presented in Fig. 7 (compare to the biological counterpart in Fig. 1). Similar to their biological prototypes the two KSs largely affect disjoint sets of joints. The joints mainly responsible for the movement in x -direction are orthogonal to the joints mainly responsible for the z -direction. Note that the human muscle-skeleton system exhibits, although more complex, a similar structure. This orthogonality suggests to combine the two KSs linearly which is done by summing up the initial posture and the two joint angle offsets $\mathbf{q}_L = \mathbf{q}_{init} + \Delta \mathbf{q}_x + \Delta \mathbf{q}_z$.

In order to show the validity of the linear superposition of the two KSs, we evaluated empirically the deviation of the actual pCoM $\langle \mathbf{f}_x(\mathbf{q}_L), \mathbf{f}_z(\mathbf{q}_L) \rangle$ from the case of perfect linear superposition $\langle \mathbf{f}_x(\mathbf{q}_{init} + \Delta \mathbf{q}_x), \mathbf{f}_z(\mathbf{q}_{init} + \Delta \mathbf{q}_z) \rangle$. The deviations for the whole support polygon can be seen in Fig. 8. Except for extremal cases, where the pCoM is located

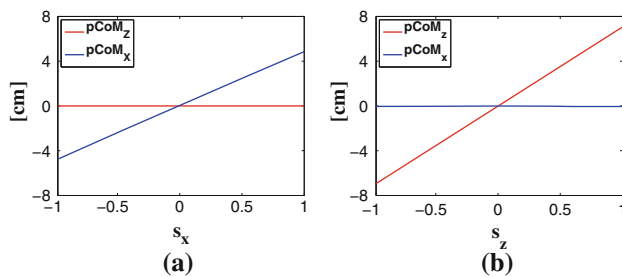


Fig. 6 **a** The plot shows the mapping of the KS-parameter s_x to the outputs $y_x = \text{pCoM}_x$ and $y_z = \text{pCoM}_z$ for the KS Φ_x . While the relationship between s_x and y_x is linear (as demanded by the definition of a linearizing kinematic synergy), y_z is nearly unaffected by s_x . **b** The same plot for the second KS-parameter s_z

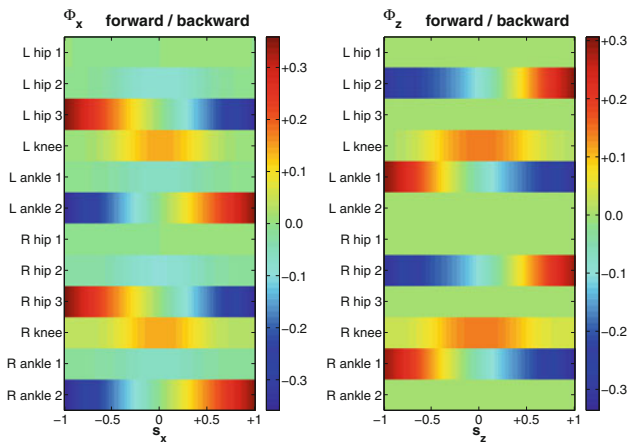


Fig. 7 Graphical representation of the KSs Φ_x and Φ_z (compare to the biological counterpart in Fig. 1). Shown are the joint angle offsets (in color coding) for the kinematic synergies Φ_x (moves the pCoM forward/backward) and Φ_z (moves the pCoM left/right) for the HOAP-2 over the range $[-1, +1]$ for the KS-parameters s_x and s_z . Note that these two KSs affect largely disjoint sets of joints

at a corner of the support polygon, the deviations from linearity are quite small.

Note that the described optimization procedure is closely related to standard JPI approaches. However, these approaches are typically used for online control, involving computationally expensive real-time calculations. With the use of kinematic synergies most of this computational load can be transferred to the offline optimization scheme. As a consequence, and as we will demonstrate later, without a significant loss of performance the robot can be balanced with very little computational power.

3.2 From statics to dynamics by using linear controllers

The kinematic synergies Φ_x and Φ_z were constructed using the pCoM as output function and, therefore, they were based on the static model of the robot. However, the robot can only

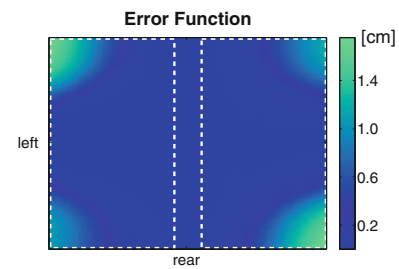


Fig. 8 Empirical evaluation of the validity of the linear superposition of the KSs Φ_x and Φ_z . We calculated the deviation of the actual pCoM $\langle \mathbf{f}_x(\mathbf{q}_{\text{init}} + \mathbf{q}_x + \mathbf{q}_z), \mathbf{f}_z(\mathbf{q}_{\text{init}} + \Delta \mathbf{q}_x + \Delta \mathbf{q}_z) \rangle$ from the case of perfect linear superposition $\langle \mathbf{f}_x(\mathbf{q}_{\text{init}} + \Delta \mathbf{q}_x), \mathbf{f}_z(\mathbf{q}_{\text{init}} + \Delta \mathbf{q}_z) \rangle$. The Euclidean norm of the deviations is shown in color code for the whole support polygon. Except for extremal cases, where the pCoM is located at a corner of the support polygon, the deviations from linearity are quite small. The white dotted lines depict the contours of the feet

estimate the mCoP with its pressure sensors⁶, which is also affected by the dynamics of the robot. Nevertheless, we are still able to use the obtained KSs in a dynamic context if following assumption holds:

Assumption: The robot moves sufficiently slowly such that

$$\text{mCoP} \approx \text{pCoM}.$$

As we will demonstrate in this section, the assumption allows us to use simple linear controllers in conjunction with the KSs. Due to the assumption, we are in principle limited to “sufficiently slow” movements. However, we will demonstrate in our experiments that a wide range of unknown external forces can be counterbalanced by our approach, despite this limitation.

We now explain how the kinematic synergy Φ_x can be used in combination with a linear controller for balancing the robot in x -direction (forward/backward). For the other KS Φ_z the process is similar. As long as the assumption holds, the function from the time derivative \dot{s}_x of the KS-parameter to mCoP_x can be approximated by a linear transfer function

$$P(z) = \frac{K}{(z - 1)}, \quad (5)$$

with $K \in \mathbb{R}^+$ and with z being the time shift operator for discrete systems (Oppenheim and Willsky 1992). The denominator polynomial represents an integrator (one pole at $z = +1$), which integrates the velocity \dot{s}_x of the KS-parameter to obtain s_x .

As long as the dynamical effects are small enough, they can be seen as uncertainties in the linear model of Eq. 5. Already a simple linear feedback controller can handle these small uncertainties. In order to obtain a closed control loop,

⁶ In our simulations of the HOAP-2, we also used simulated pressure sensors to calculate the mCoP.

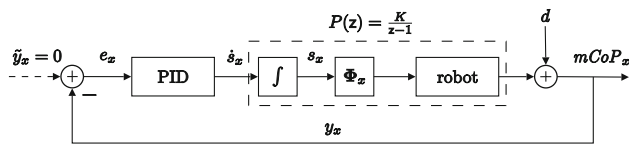


Fig. 9 Closed control loop for the kinematic synergy Φ_x . Since we want to have the mCoP_x at the center of the support polygon, the reference point is set to $\tilde{y}_x = 0$. The external perturbation d results from external forces and/or model uncertainties

we define a feedback error

$$e_x := \tilde{y}_x - y_x \quad (6)$$

with \tilde{y}_x being the desired output value and $y_x = \text{mCoP}_x$. The goal is to prevent the robot from falling. Therefore, the mCoP_x should stay close to the center of the support polygon. Since we have defined the center of SP at the origin, see Fig. 3c, the desired value \tilde{y}_x is set to 0.

We can now use a general standard PID controller to get the controller output \dot{s}_x

$$\dot{s}_x = K_P e_x + K_I \int e_x dt + K_D \frac{de_x}{dt}, \quad (7)$$

where K_P , K_I , and K_D are the positive PID controller parameters. Figure 9 shows the described closed control loop for the kinematic synergy Φ_x . Since the plant (see Eq. 5) already contains an integrator, the use of PD controllers ($K_I = 0$) is sufficient. For the KS Φ_z , we used a similar control loop, which worked independently from and in parallel to the first control loop.

We have described the control scheme to control around a set point ($\tilde{y}_x = \tilde{y}_z = 0$). However, the control loop can also be used to move the mCoP on any desired time-varying trajectory⁷, i.e., $\tilde{y}_x(t)$ and $\tilde{y}_z(t)$. This is useful in many applications. For example, for the purpose of initiating a walking cycle, the robot has to move its mCoP under the future supporting foot in order to be able to raise the other leg without falling.

The controller parameters used in the experiments were empirically found to have a reasonable performance. As we demonstrate (see Sect. 4.3) there is a wide range of appropriate controller parameters and, therefore, the choice of the parameters is not critical.

Linear and non-linear control theory offers a number of possible improvements for the controllers, for example, adaptive control (see Åström and Wittenmark 1995) or robust control schemes, optimal control, and different trial and error approaches to find good control parameters (see for example Kuo and Golnaraghi 2002). Even higher order controllers or different control structures than in Fig. 9 could be used. However, in order to illustrate the capability of using kinematic

synergies for balance control, we only use the previously presented, simple PID controllers.

3.3 Examination of different possible perturbations

Lets take a closer look at possible perturbations d for the proposed control loop (Fig. 9). We will distinguish between three different kinds of perturbations:

1. **Model perturbations:** Since we obtained our KSs from the static model of the robot, unmodeled dynamics, which will always be present to some extent, result in model perturbations.
2. **Internal perturbations:** The mCoP is also influenced by movements of joints, which are not under the control of the kinematic synergies. For example, if our humanoid robot uses the presented KSs for balancing and additionally moves a heavy weight with its arms, this movement will also change the mCoP position. Note that the proposed control loop does not need any information about the movements of these joints.
3. **External perturbations:** For example pushes, pulls, contact with the environment, or a moving support platform.

Since a standard feedback control loop has the property to suppress the perturbations d , our approach works for a wide range of tasks. As shown in our experiments (Hauser et al. 2007), these tasks include counteracting external forces, following trajectories, compensating for forces introduced by movements of the limbs of the robot or even a mixture of these tasks. If the perturbation is too large, the assumption ($\text{mCoP} \approx \text{pCoM}$) might be violated and the controller will, therefore, not be able to compensate the resulting error anymore. Yet, as our experiments show, the proposed system is capable to react appropriately to a wide range of perturbations.

4 Experiments

We conducted experiments with our proposed approach for a variety of possible applications. We demonstrate that kinematic synergies with linear controllers empower a humanoid robot to counterbalance different kinds of dynamic perturbations. In our first experiments, the robot had to counteract a moving support surface (platform where it stood on) and abrupt unforeseen external forces at the same time (see Sect. 4.1). Subsequently, we show that the approach can also be extended easily to balancing in single support (the robot only stood on the left foot, see Sect. 4.2) and that robustness against parameter changes is an inherent property (Sect. 4.3). Furthermore, we compare our approach to an online Jacobian Pseudo-Inverse (JPI) algorithm. Finally, we dem-

⁷ We have already demonstrated that in Hauser et al. (2007).

onstrate that our approach can be easily transferred from the simulation to the real robot without any special precautions (Sect. 4.5).

All simulations were implemented in the robot simulation software Webots, see Michel (2004). A detailed model of the dynamics of the HOAP-2 robot, based on data provided by the vendor Fujitsu, was used. The basic simulation time step was set to 2 ms and the time steps for the control loops were set to 8 ms. In the general setup, we had two kinematic synergies (Φ_x and Φ_z), which were used within two separate control loops. They reacted independently from each other on their corresponding output dimension x and z . In dependence on their errors e_x and e_z , both linear controllers calculated the velocities \dot{s}_x and \dot{s}_z of their KS-parameters. The velocities were integrated numerically to obtain s_x and s_z , which were then mapped via the look-up table into joint angle offsets. Subsequently, these joint angle offsets were linearly combined as described in Sect. 3.1 to get the actual joint target angles. Finally, these angles were transformed into torques by local PD controllers⁸ at the servos.

We provide supplementary multimedia material in form of two videos. The first one (*simulation_videos.avi*) shows all simulated experiments of the following sections. The second video (*real_robot_videos.avi*) shows the experiments with the real robot. Both videos (in compressed form) are about 13 MB in size.

4.1 Moving support platform (surfboard task)

In this task, we simulated the HOAP-2 robot standing on a movable support platform (surfboard). The surfboard could rotate about the x -axis with angle Θ_x and about the z -axis with the angle Θ_z . Typical scenarios of the setup can be seen in Fig. 10.

We considered the case where the surfboard was tilted dynamically in random directions. The random trajectories for the angles Θ_x and Θ_z were generated independently from each other by smoothing (by the use of a discrete low-pass FIR-filter⁹) random trajectories of jumps (steps) with random amplitudes and random durations. Typical resulting trajectories are presented in Fig. 11a and b.

In addition to the random movement of the surfboard, unforeseen external forces (for example these forces could arise from wind or contact with other objects) were applied to the torso of the robot at various points in time. We designed this scenario in order to show that our proposed approach is able to deal with different kinds of external perturbations simultaneously. Furthermore, control strategies that require knowledge of the dynamic model of the robot are

⁸ Note that these are the hardware controller of the servos and not the controllers from our proposed control loops.

⁹ The used FIR-filter had three poles at 0.997.

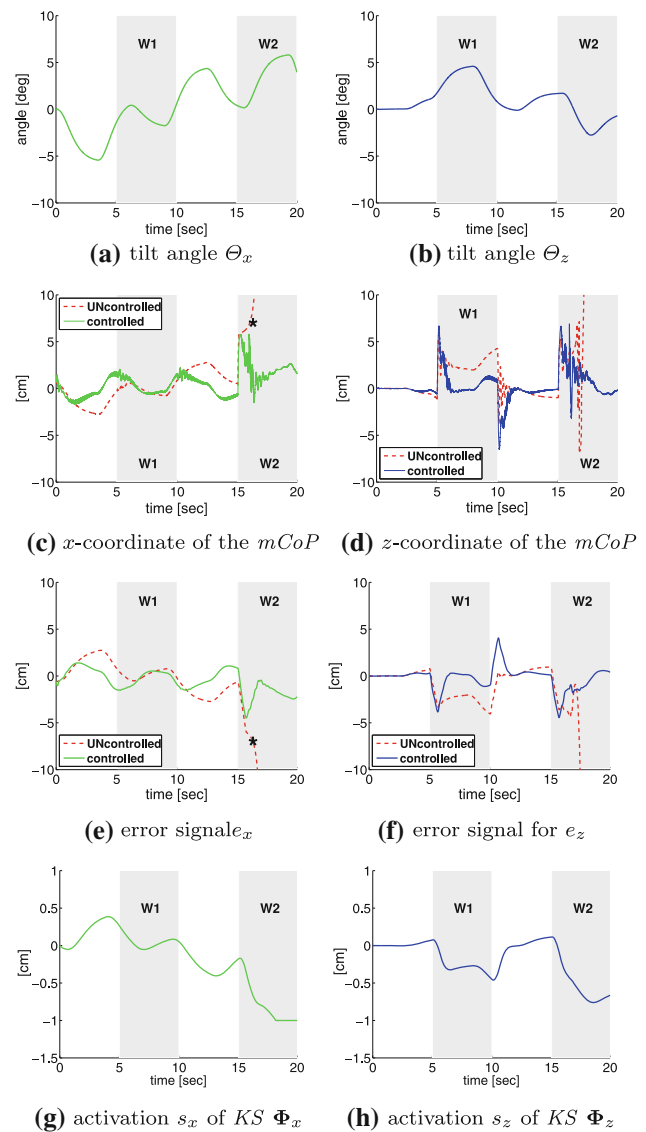


Fig. 10 Screenshots of the posture of the (simulated) HOAP-2 at 4 time points during the balancing experiment with the random moving support surface (surfboard) and external perturbations (winds). In **b**, the wind $W1$ was blowing from the right (point of view of the robot; red arrow). As a consequence, the robot was leaning against the wind in order to move its mCoP back into the middle of the support polygon. In **d** another wind $W2$ was blowing from the right and the back (red arrow), resulting in a diagonal force. Again, the robot responded properly to this online modification of its dynamic model

inapplicable in this scenario, because the external forces change the dynamic model of the robot in an unknown, online manner. Figure 11 shows the results when an external force $W1 = [0, 0, 5]^T N$ (a force from the right side) was applied at the torso of the robot during the interval [5 s, 10 s], and another external force $W2 = [5, 0, -5]^T N$ (a force from the right and the back) was applied during the interval [15 s, 20 s] (we shaded these two time intervals in gray). Note that the onsets of the winds were abrupt (i.e., a step function in time)

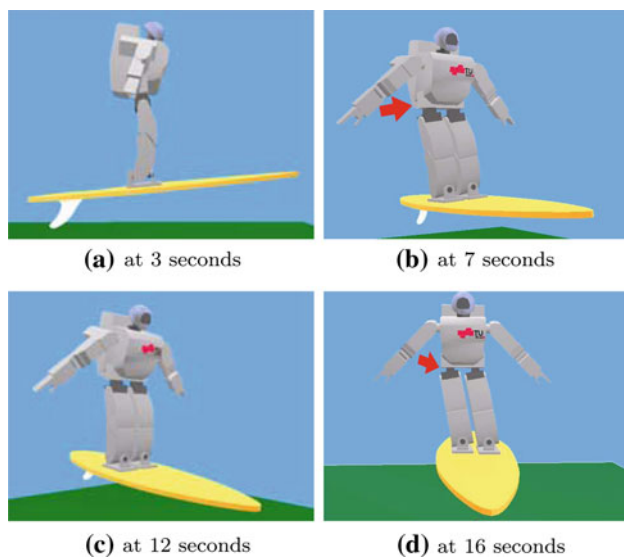


Fig. 11 Result for the experiment with a moving support platform (surfboard) and unexpected external forces (wind) W1 and W2. The balance of the HOAP-2 is controlled by two linear controllers combined with the kinematic synergies. Without balance control (red-dashed line in **c** and **d**) the mCoP left the support polygon after 16s (in response to the wind W2), and the robot fell over. With balance control (solid lines) the stability of the robot was maintained in spite of unexpected external forces

and, therefore, represented highly dynamical perturbations to the system.

Typical trajectories of the mCoP for the described setup, with and without balance control, are shown in Fig. 11c and d. Without balance control, the robot lost balance after 16s (indicated by a black star in Fig. 11c and e), whereas with our controllers, balance was maintained. The error signals for both dimensions x and z can be seen in Fig. 11e and f. Note that both perturbations, the movements of the surfboard and the external forces, are *external perturbations*. In addition, as the setup was dynamic, inherent *model perturbations* were present too. With this experiment, we demonstrated that our approach is able to react online against a mixture of different types of unforeseen perturbations.

4.2 Kinematic synergies in single support

In this experiment, we demonstrate how to apply our approach in single support. We used two different strategies. The first strategy reused the KSs previously calculated for double support (referred to as DS-KS). We switched off the output of the control loop for the joints of the lifted leg and set the desired mCoP position to the center of the reduced support polygon (defined by the single supporting foot). The second strategy was to design new KSs for single support (referred to as SS-KS). We used the same procedure as described previously in Sect. 3.1, with the distinction, that we used a

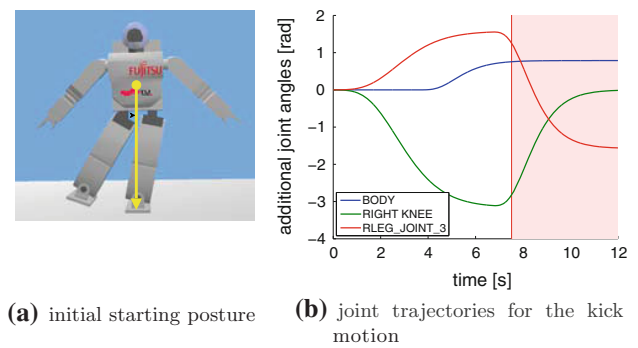


Fig. 12 Setup for the single support task. **a** Shows the initial posture. The yellow circle denotes the CoM of the robot and the arrow points to the pCoM, which is located at the center of the support polygon. **b** Shows the joint angle trajectories which were used for the kick motion. When no balance control was applied, the robot lost balance and tipped over at about 7.5 s

different initial position (the one shown in Fig. 12a) and we only optimized the joint angles of the supporting leg.

In the experimental setup, the robot stood only on its left foot. The right foot had no contact to the ground and therefore the right leg was free to perform any desirable movement, for example, a kick motion. The initial posture can be seen in Fig. 12a. The corresponding s -values for this posture were $s_x = 0$ and $s_z = 0.195$ for DS-KS and $s_x = s_z = 0$ for SS-KS.

In order to demonstrate the validity of both strategies, we moved the body joint and the hip joints of the left leg (these joints were not under the control of the KSs) in order to perform a kick motion, which also included the upper trunk (see Fig. 12b). For the robot this movement represented an *internal perturbation* as discussed in Sect. 3.3. When no balancing control was active, after about 7.5 s of simulation time, the robot tipped over and fell. With the controllers switched on, the robot was able to keep balance during the kick motion (in both cases, SS-KS and DS-KS). Figure 13 shows the time course from 2 to 12 s of this experiment with DS-KS. Similar results were obtained with SS-KS. Figure 14a and b shows the trajectories of the KS-parameters s_x and s_z . Note that in the case of DS-KS, there was an offset at the beginning of the simulation for the KS-parameter s_z . This reflects the offset of the initial posture for single support from the original initial posture for double support. Figure 14c and d presents the errors during the simulation. The controllers counteracted the disturbances correctly and kept the errors close to zero for both strategies. The dashed red curve shows the errors when no controllers were activated. Note that the scales of the y -axes of the plots in Fig. 14 are different for the dimensions x and z . This is a consequence of the used kick motion which mostly affected the mCoP in the x direction (forward/backward). Both strategies (DS-KS and SS-KS) showed a similar performance (see Fig. 14c, d). As a consequence, we can see that the KSs can also be used for different, albeit related tasks,

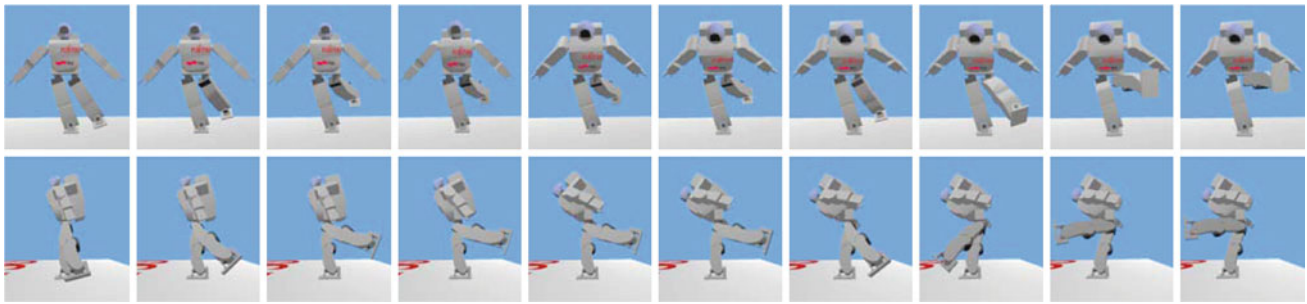


Fig. 13 Postures of the simulated HOAP-2 for the single support task. The *first* row shows the robot from the front and the *second* row shows the robot from the side. Screenshots were taken every second from the 2nd to the 11th second

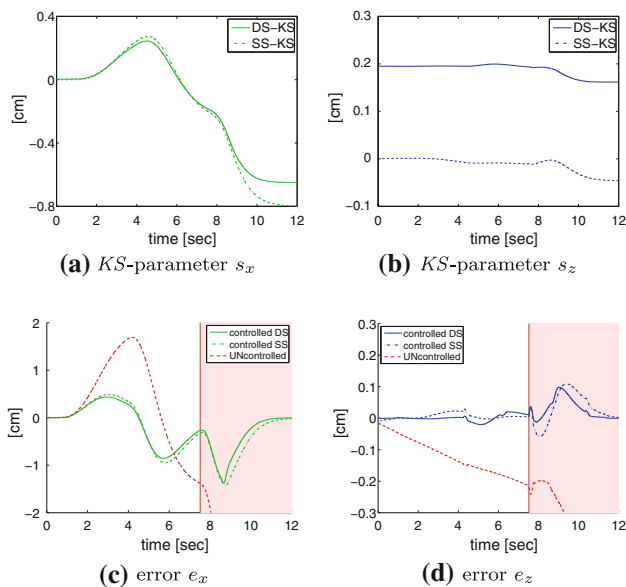


Fig. 14 Results for the single support task. The *left* column shows the results for the x -dimension (Φ_x , forward/backward) and the *right* column the results for the z -dimension (Φ_z , left/right). **a** and **b** The responses of the KS-parameter s_x and s_z for both approaches (SS-KS and DS-KS). **c** and **d** The errors. The *red-dashed* curves denote the errors when no balance control was active. In this case, the beginning of the *red* region indicates, when the robot tipped over and lost balance

for which, in the first place, they have not been designed for. This might also help to reduce the number of needed KSs in real world applications, because related tasks might share the same set of KSs.

4.3 Robustness to changes in the model of the robot and the controller parameters

The kinematic synergies are based on the static model of the robot. Since uncertainties in the model parameters (lengths and masses) are common, it is desirable to have a framework that is robust to changes in those parameters. Moreover, such a robustness simplifies a transfer from the simulation to a real robot. In addition, it would be beneficial to have a wide

range of valid control parameters, i.e., K_P and K_D , which are able to balance the robot. In the following experiments, we demonstrate that our proposed setup is widely robust to variations of these parameters.

In a first experiment, we varied the size of the robot by changing the length of every link by a multiplicative *length factor*, ranging from 0.5 to 2.5. We used the trajectory following task from our previous work [Hauser et al. \(2007\)](#), where the robot had to follow a figure eight trajectory with its mCoP, while it manipulated a heavy weight. The KSs were kept constant. First, we used the same controller parameters for both controllers as in the original task ($K_P = 80$ and $K_D = 0.1$). The robot was able to keep balance for a *length factor*, which ranged from 0.85 to 1.1. In [Fig. 15](#), the mean squared errors for the x -dimension¹⁰ for successful *length factors* (the robot kept balance) are indicated by red circles for these controller parameters. In order to demonstrate how to improve robustness, we increased the response time of the controllers by setting the controller parameters to $K_P = 50$ and $K_D = 0.0$. In this case, successful *length factors* ranged from 0.85 to 1.45 (indicated by blue crosses in [Fig. 15](#)). Note that the mean squared error only increased slightly. We also tested an even slower controller ($K_P = 20$ and $K_D = 0.0$), which resulted in a fairly large range from 0.7 to 2.25 (indicated by green triangles in [Fig. 15](#)). However, the used controller was too slow to follow the desired trajectory, which can be seen in the high mean squared error values. The corresponding mCoP trajectories of all three controllers can be seen in the right plots of [Fig. 15](#). The black lines are the target trajectories. The conclusion of the experiment is that the proposed setup is robust to changes in the lengths of the robot. In addition, the results suggest that there is a tradeoff between the robustness of the approach and the response times of the controllers. Similar results were obtained, when the masses as well the lengths were changed simultaneously to simulate growing.

In a second experiment, we provide an evaluation of the robustness of our approach to the choice of the controller

¹⁰ Similar plots were obtained for the z -dimension.

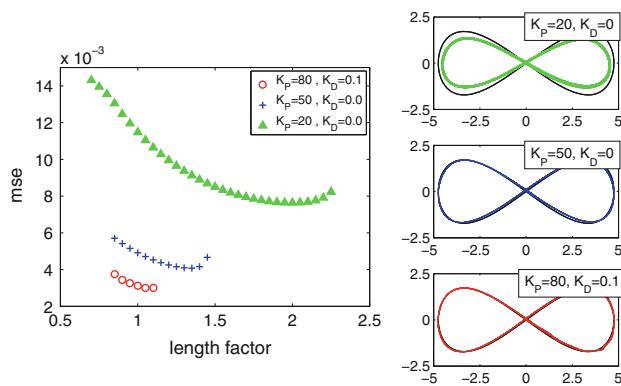


Fig. 15 Results on robustness to changes in the lengths. The lengths of all links were multiplied by a *length factor*. The plot shows three different settings for the controller parameters, resulting in different response times of the controllers. The red circles show the mean squared error (mse) for the controller ($K_P = 80$ and $K_D = 0.1$) with the shortest response time. The red circles are shown for the range of successful *length factors* (the robot kept balance) from 0.85 to 1.1. By increasing the response time (controller parameters were set to $K_P = 50$ and $K_D = 0.0$) the range (from 0.85 to 1.45) of successful *length factors*, and therefore, the robustness of our approach could be increased. However, also the mse increased slightly, which indicates a worse tracking performance. With an even longer response time ($K_P = 20$ and $K_D = 0.0$), the region of successful *length factors* (from 0.7 to 2.25) also grew, however, the controller was no longer able to follow the desired trajectory (indicated by the large mse values). The results point to the fact, that there is a tradeoff between the robustness of the approach and the response time of the controller. The right plots show the corresponding mCoP trajectories for the three controllers (at a length factor = 1)

parameters. We used the single support task described in Sect. 4.2 (using the previously described SS-KS) and varied the K_P and K_D parameters over several decades. We evaluated which parameter settings (K_P/K_D -pairs) were successful, i.e., the robot was able to keep balance. The results can be seen in Fig. 16. Successful parameter settings are highlighted in green. Note that the region of successful settings ranges over two decades for both parameters. This suggests that our approach is robust to the choice of the controller parameters and, thus, appropriate parameters are easily found. Moreover, this robustness potentially allows us to combine our approach with adaptive control as in Åström and Wittenmark (1995) or online policy search methods (Kober and Peters 2009).

4.4 Comparison to an online Jacobian pseudo-inverse approach

As we have already mentioned in the introduction, our main motivation was to adopt the biological concept of synergies, rather than outperforming existing approaches. Nevertheless, it is interesting to see how our biologically inspired approach performs compared to pure algorithmic approaches, like an online Jacobian Pseudo-Inverse (JPI) approach (Sciavicco and Siciliano 1999). This approach performed online an

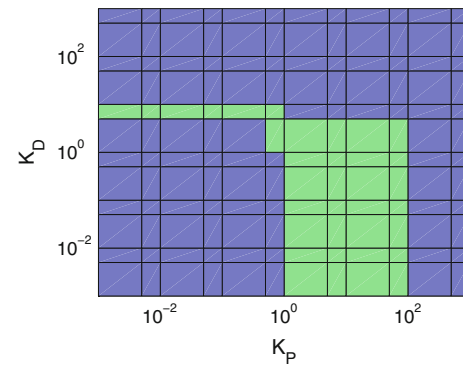


Fig. 16 Region of evaluated controller parameters for the single support task described in 4.2. Successful parameter settings (for which the robot was able to keep balance) are highlighted in green. Note that the scales of the axes are logarithmic. The region of successful controller parameters ranges over two decades for both parameters, indicating that our approach is robust to the choice of the control parameters

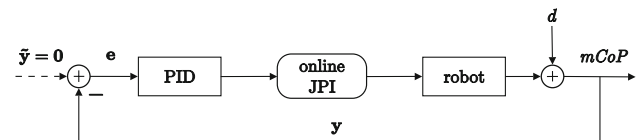


Fig. 17 Schematic setup of the online Jacobian Pseudo-Inverse (JPI) approach, to which we compared our approach (Fig. 9). Instead of fixed kinematic synergies, this approach has to run online an optimization process (based on a JPI) at every single time step to calculate the optimal joint angles velocities

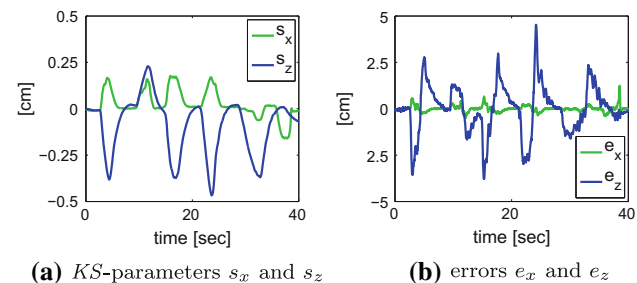


Fig. 18 Resulting responses of the HOAP-2 to external forces. The screenshots were made during dynamic action. The top row shows screenshots for experiments while standing on the floor (setup F). External forces were applied by pushes. The second row shows screenshots of experiments with the robot standing on a movable platform (setup P). External forces were applied by moving the platform. In any of these situations, the robot acted correctly and moved its mCoP to the desired position at the center of the support polygon. Note that there are videos of the experiments available as supplementary material

optimization similar to the one we used for the offline construction of the KSs. In order to be responsive to external perturbations and model uncertainties, we had to plug the JPI into a feedback control loop.

Figure 17 shows the considered setup. In order to compare both approaches, the robot had to track a rectangular trajectory (with rounded edges) centered at the center of



Fig. 19 The KS-parameters and the errors signals recorded during an experiment with the real HOAP-2 robot. The robots were pushed from different directions (setup **F**). The *left figure* shows the KS-parameters and the *right figure* shows the corresponding error signals. We can see

that, except for a short time period after a change of the applied external force, the error is kept close to zero. This indicates that the robot always tried to maintain its mCoP at the center of the support polygon

the support polygon. We systematically increased the size of the rectangle and the speed of the trajectory and compared the maximum quantities, at which the robot tipped over. The differences between the two approaches for both limits (rectangle size and speed) were less than 1%. Hence, there was no significant difference in their performances. This suggests that the complex Jacobian pseudo-inverse computations can be performed offline (in order to construct the KSs) without a significant loss of performance. Note that the JPI approach needs to apply online sophisticated, time intensive calculations, while our approach is based on a much simpler control law using only a PID controller. A comparison of the online computation time of both approaches revealed a speed-up factor of 80 in favor of our approach. The results also show that the performance loss due to the linear superposition¹¹ of the two KSs is negligible for humanoid balancing.

4.5 Experiments with a real HOAP-2 robot

In our final experiment, we transferred our approach to a real HOAP-2 robot. Due to the previously demonstrated robustness against model uncertainties, we were able to simply reuse the same KSs as in our simulations, even though the static model used for the KSs did not perfectly match the static model of the real robot.

We investigated two different setups. In the first setup, the robot stood on the floor (denoted by **F**) and we applied external forces. This was done by applying an almost constant force from different directions for approximately 1 to 2 s by pushing the robot. In the second setup (denoted by **P**),

we reproduced the surfboard task. The robot stood on a movable platform, which was mounted on a plastic sphere in order to resemble the surfboard with its two degrees of freedom. In contrast to the simulated experiment, no additional external forces (winds) were used (only the movement of the platform represented an external force). Note that in both setups, the robot had no knowledge about the onset times, the directions, or the amplitudes of the applied external forces.

The first row of Fig. 18 shows the responses of the robot to pushes from different directions (setup **F**). The second row shows responses of the robot to different movements of the supporting platform (setup **P**). The robot counterbalanced the applied external forces in order to keep its mCoP at the middle of the support polygon in each of these cases.

In Fig. 19, we show typical KS-parameters and the error signals recorded while the robot was pushed from different directions (in setup **F**). Note that except for a short time period after a change of the applied external force, the error was kept close to zero. This indicates that the robot always tried to maintain its mCoP at the center of the support polygon. Note that videos of the experiments are provided in the additional multimedia file.

5 Conclusion

We have presented a new approach toward balance control of a humanoid robot that was based on inspiration from biology. We resembled the concept of muscle synergies in physiology by introducing kinematic synergies based on the coordination of joint angles. Our main motivation was to adopt the biological concept for robots rather than to outperform existing approaches. However, as we have shown our biologically inspired approach performs comparable to the standard approach, i.e., JPI. For the special case of balance control of the humanoid robot HOAP-2, we demonstrated how such

¹¹ Note that the JPI approach does not use a linear superpositions, but rather simultaneously optimize for both output dimensions, i.e., $y \in \mathbb{R}^2$.

KSs can be constructed in such a way that their superposition is almost linear (like in biological paradigms), although each KS itself is highly non-linear. Based on this concept, we were able to demonstrate that it is possible to move the time intensive calculations of the optimization process offline and therefore keep the needed online calculations simple and fast. We have demonstrated, both through computer simulations and through experiments with the real robot HOAP-2, that this strategy makes it possible to virtually reduce the highly non-linear balance control problem of the robot to a linear control problem (as long as the required movements are not too fast). Note that, although it is well established by experimental data that biological synergies exist, so far it was not known whether they provide some tangible advantage for motor control or not. Our results suggest that they enable biological organism to linearize certain non-linear motor control problems.

We also showed that, in contrast to approaches, which are based on an exact dynamic model of the robot, our proposed combination of KSs and linear controllers enables a humanoid robot to counterbalance unknown external forces of different kinds. Additionally, we demonstrated that robustness to parameter changes in the model as well to changes in the controller parameters is an inherent property of the proposed approach. Based on this robustness, we were able to transfer in straightforward manner this new approach for balance control from a simulated to a real HOAP-2 robot.

Acknowledgements Written under partial support by the Austrian Science Fund FWF project # P17229-N04.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Alexandrov A, Frolov A, Massion J (1998) Axial synergies during human upper trunk bending. *Exp Brain Res* 118(2):210–220
- Åström KJ, Wittenmark B (1995) Adaptive control. In: Adaptive control, 2nd edn. Addison Wesley Longman, Reading
- Baerlocher P, Boulic R (1998) Task-priority formulations for the kinematic control of highly redundant articulated structures. In: Proceedings of the 1998 IEEE/RSJ international conference on intelligent robots and systems, vol 1, pp 323–329, 13–17 October 1998
- Baerlocher P, Boulic R (2004) An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *Vis Comput* 20(6):402–417
- d'Avella A, Bizzi E (2005) Shared and specific muscle synergies in natural motor behaviors. *Proc Natl Acad Sci* 102(3):3076–3081
- d'Avella A, Saltiel P, Bizzi E (2003) Combinations of muscle synergies in the construction of a natural motor behavior. *Nature* 6(3):300–308
- Freitas SMSF, Duarte M, Latash ML (2006) Two kinematic synergies in voluntary whole-body movements during standing. *J Neurophysiol* 95(2):636–645
- Goswami A (1999) Postural stability of biped robots and the foot rotation indicator (FRI) point. *Int J Robot Res* 18(6):523–533
- Goswami A, Kallem V (2004) Rate of change of angular momentum and balance maintenance of biped robot. In: Proceedings of the 2004 IEEE international conference on robotics and automation ICRA, vol 4, pp 3785–3790, April 2004
- Hauser H, Neumann G, Ijspeert AJ, Maass W (2007) Biologically inspired kinematic synergies provide a new paradigm for balance control of humanoid robots. In: Proceedings of the 7th IEEE RAS/RSJ conference on humanoid robots (HUMANOIDS07), Pittsburgh, PA, December 2007
- Kagami S, Kanehiro F, Tamiya Y, Inaba M, Inoue H (2001) Auto-Balancer: an online dynamic balance compensation scheme for humanoid robots. In: Donald BR, Lynch K, Rus D (eds) Algorithmic and computational robotics: new directions. A K Peters Ltd., Wellesley, pp 329–340
- Kajita S, Yamaura T, Kobayashi A (1992) Dynamic walking control of a biped robot along a potential energy conserving orbit. *IEEE Trans Robot Autom* 8(4):431–438. ISSN 1042-296X
- Kajita S, Kanehiro F, Kaneko K, Yokoi K, Hirukawa H (2001) The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: Proceedings of the 2001 IEEE/RSJ international conference on intelligent robots and systems, Maui, pp 239–246
- Kober J, Peters J (2009) Policy search for motor primitives in robotics. In: Advances in neural information processing systems 22 (NIPS 2008). MIT Press, Cambridge, pp 849–856
- Kuo BC, Golnaraghi F (2002) Automatic control systems, 8th edn. Wiley Inc., New York
- Lee S-H, Goswami A (2007) Reaction mass pendulum (rmp): an explicit model for centroidal angular momentum of humanoid robots. In: IEEE international conference on robotics and automation, 10–14 April 2007, pp 4667–4672
- Mahboobin A, Loughlin PJ, Redfern MS, Anderson SO, Atkeson CG, Hodgins JK (2008) Sensory adaptation in human balance control: lessons for biomimetic robotic bipeds. *Neural Netw* 21(4): 621–627. ISSN 0893-6080. Robotics and Neuroscience.
- Mansard N, Chaumette F (2007) Task sequencing for high-level sensor-based control. *IEEE Trans Robot* 23(1):60–72. ISSN 1552-3098
- Mason CR, Gomez JE, Ebner TJ (2001) Hand synergies during reach-to-grasp. *J Neurophysiol* 86(6):2896–2910
- Michel O (2004) Webots: professional mobile robot simulation. *J Adv Robot Syst* 1:39–42
- Mussa-Ivaldi FA (1999) Modular features of motor control and learning. *Curr Opin Neurobiol* 9:713–717
- Oppenheim AV, Willsky AS (1992) Signal and systems. Prentice-Hall Inc., Englewood Cliffs
- Peterka RJ (2009) Comparison of human and humanoid robot control of upright stance. *J Physiol Paris* 103(3–5):149–158. ISSN 0928-4257. Neurorobotics
- Popović MB, Goswami A, Herr H (2005) Ground reference points in legged locomotion: definitions, biological trajectories and control implications. *Int J Robot Res* 24(12):1013–1032
- Sabatini AM (2002) Identification of neuromuscular synergies in natural upper-arm movements. *Biol Cybern* 86(4):253–262
- Sciavicco L, Siciliano B (1999) Modelling and control of robot manipulators, 2nd edn. Springer, Berlin
- Torres-Oviedo G, Ting LH (2007) Muscle synergies characterizing human postural responses. *J Neurophysiol* 98(4):2144–2156
- Tricon V, Le Pellec-Muller A, Martin N, Mesure S, Azulay J-P, Vernazza-Martin S (2007) Balance control and adaptation of kinematic synergy in aging adults during forward trunk bending. *Neurosci Lett* 415(1):81–86

- Vukobratović M, Borovac B (2004) Zero-moment point—Thirty five years of its life. *Int J Humanoid Robot* 1:157–173
- Wang Y, Zatsiorsky VM, Latash ML (2005) Muscle synergies involved in shifting the center of pressure while making a first step. *Exp Brain Res* 167(2):196–210
- Whitney DE (1969) Resolved motion rate control of manipulators and human prostheses. *IEEE Trans Man Mach Syst* 10(2):47–53. ISSN 0536-1540